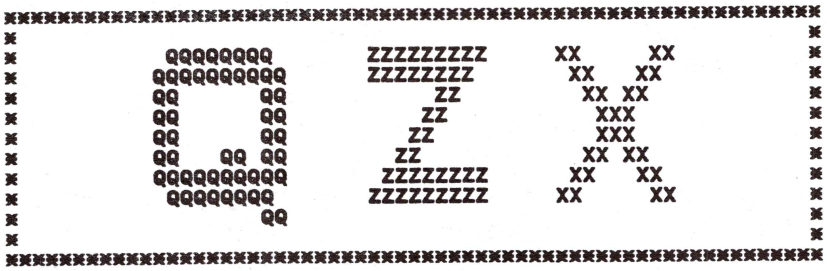


The Journal Covering Amateur Radio & Sinclair Computers
ZX-80; Micro-Ace; ZX81; T/S 1000, 1500, & 2068
2025 O'Donnell Drive
Las Cruces, New Mexico
88001

K5XY
Alex. F. Burr
Publisher

W4GHV-5
Bo Barry
Editor

Volume 4 Number 5 *($1.50-issue)* *April 1986*

* *Putting the Timex/Sinclair on Packet Radio, Part II (+1/2), W4GHV*
* *Serial Interface Project for the ZX81/TS1000, Part II (from CTM)*
* *Publisher's Notes, K5XY*
* *Split Screen Scroll Routine, WA0MRG*
* *Using Extra Keys on the TI Keyboard, LU8MAD*
* *Evaluating Color Monitors For Use With the 2068, WA6DLI*
* *Bits and Bytes*
* *QZX Friends*
* *Products of Special Interest*
* *Unclassified Ads*
* *Equipment Sources for your Timex/Sinclair*

COMING UP:
Controlling your Transceiver (FT-757GX) From Your TS-1000

32K BANKED MEMORY

Plugs into the TS2068 cartridge slot. This 32K ram is much more than just a memory. An on board battery keeps it alive even when you turn the computer off. Switch selectable for use in the DOCK or EXROM banks with NO mods to the computer. Write protect switch lets you use memory like an EPROM. Run your own plug-in BASIC programs! Extend capacity by 32K. Reduce or eliminate tape loading time. Detailed instructions include utilities for bank switching and data transfers. The perfect tool for extending your 2068's memory or for writing and debugging EPROM software. A beautiful board: solder masked, gold tipped, satellite grade anti-static coating. Very dependable and rugged. Battery included. Only $109.95

EXTENDED BASIC

22 new commands for your TS1000! Advanced screen utilities make your ZX81 look more like a monochrome 2068. Includes READ, DATA, RESTORE, FILL MOVE, LEFT$, MID$, RIGHT$. IN and OUT commands give you Basic control of I/O mapped peripherals which were previously accessed only in machine code. Extended Basic does not require the use of PEEK, POKE, or USR. Takes up just 3.5K of memory. 24 page manual, sample listings included. Unbelievable speed, excellent documentation. NO USR CALLS!! Only $19.95

WATOR • ZX81!

Now the ZX81/TS1000/1500 has its very own (and very excellent) version of this popular ECOLOGY SIMULATION program; the idea of which first appeared in Oct. '84 Scientific American. WATOR recreates the ecology of 3 interdependent species. You can experiment by changing populations, feed cycles, reproduction, etc. Watch generations pass on the TV screen. Generate HI-RES population curves on the TS2040 printer. Very useful for studying predator/prey relationships. On cassette: Just $16.95

UNIVERSAL I/O PORT

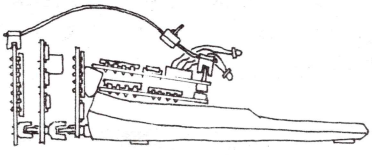
The ideal peripheral for experimentors. Learn to control and sense external devices using your computer. Works with ALL Sinclair/Timex machines. 8 bits parallel input, 8 bits latched output plus 2 handshake lines. Easy to set up or reconfigure many times. This board forms the foundation of a series of software applications which teach concepts of port programming. Choose one application listed below ABSOLUTELY FREE when you order a port board. Only $69.95 each or 2 boards for $109.95

APPLICATIONS AVAILABLE NOW!!

*Morse Code Translator for TS2068--Accepts tone decoder input, outputs ENGLISH on the TV screen or printer. Full screen display, auto scrolling. 5K buffer. Unique built-in signal reader aids in detecting/reducing noise from the system. This program is a great tool for learning morse code. Instructions cover set-up, tips for noise reduction, and a detailed tutorial on CW translation theory. Regularly: $16.95

*Pseudoscope for TS2068--A serial bit stream analyzer which mimics the action of an oscilloscope. Use to graphically portray low frequency digital signals in a horizontally scrolling graph. Program lets you store and compare samples. Print them out on a 2040 printer. Adjustable sample rate, 60hz timing dots are plotted with your input sample. Pseudoscope lets you SEE morse, rtty, serial keyboard signals, etc. as the computer sees them. Very useful in program development or in optimizing interface hardware. Tape/instruction book/full disassembly: $16.95

*Computer I/O Data Communications--Lets your 2068 speak to your ZX81! Use your TS1000 (must have 64K ram) to store data for your 2068. Send commands which transfer data back and forth, or cause the TS1000 to execute your own special Basic commands. Now 2068 owners can put their old 1000's back on line. Software includes operating systems for both computers. Requires 2 port boards (one for each computer). Tape/instructions: $16.95.



Please add $1.50 per order for postage. Phone orders are gratefully accepted.
To order, send check, money order, or credit card information to:

Thomas B. Woods
P.O. Box 64, Jefferson, NH 03583
Phone (603) 586-7734

Software & Hardware for TIMEX-SINCLAIR

Putting the Timex on Packet Radio
Part II and a half
Bo Barry, W4GHV

Running behind schedule, but slowly catching up! I made it through the semester (I'm a college prof) and now have 3 months to catch up. Other than a little communication with my Model 100 on packet radio, I haven't had much time for ANYTHING else. Jim Henderson, WB9KNF came to our rescue, providing the missing second part of Vic Rice's RS-232 article. Reading over it, I decided it was 'different enough' from the one I improvised last month, so I'm reprinting the REAL second part of the article. Keep on working, reading about packet, getting in the mood, checking to see if there is any local packet activity, etc, and I'll have more accomplished next month, hopefully.

I don't recall if I provided this before or not! Here's a few articles to dig out if you want to learn more about packet radio:

73 Mag- Sept & Oct 83, Jan 84; Ham Radio- June 79; QST-July 85; Ham Radio- July & Aug 83; CQ- Nov 85. CTM magazine appears to be the front-runner nowadays as far as packet coverage goes. A subscription to it would keep you up to date.

Corrections to the previous article(s): Pin 17 of the 8251 should be connected to pin 8 of the 1489 instead of pin 9. In the program, CONTROL (C) should refer to address 8193 and DATA (D) to 8192. Also, when you round up an 8251 be sure to get the 8251A. I had some strange problems a while back with a RS-232 interface when I tried to substitute an 8251 for a defective 8251A. I have no idea of the differences between the two chips. The 8251A works all the time.

Here's the second part to Vic Rice's article:

A Serial Interface Project
for the ZX-81 & TS/1000
Part II
(from CTM Magazine)

This is the second article of a series dealing with a serial interface with which to connect your ZX81/TS1000 to a printer, modem or other device. Part one dealt with the actual wiring of the circuit; this one will cover the programming required to obtain useful work from the device.

The real heart of the circuit is the 8251; the rest of the chips on the board exist only in a supporting role. The 8251 is a prime example of a labor saving 'black box'. You pass it data, and it will look after serializing each byte of data, adding 'start' and 'stop' bits, generating parity and then sending the resultant bit stream at some given rate of speed. For incoming data it will strip off the start & stop bits, do parity checking for you and assemble and pass over the full bytes of data as you request them. In order to do its job, however, it will require some instructions from you the programmer.

As you should have noticed in the wiring diagram, there are two 'ports'shown on the 8251. These are specifically the 'CONTROL' port and the 'DATA'port. The names are indicative of their functions: the control port is used to control the functions of the 8251, and the data port is used to pass data back and forth between the computer and the outside world. Once the 8251 has been 'initialized' (told what you want it to do), then sending or receiving data is a straightforward movement of data a byte at a time into and out of the data port. Software-wise, the interesting part is thus the control port.

When you first apply power to your Sinclair, control passes to the 'cold start' routines of the ROM. This is a set of pre-coded routines which clear and verify storage, set up the display file and generally get the computer ready for work. In the same manner as the ROM routines get the computer ready for work, so you have to supply a set of routines to get the 8251 ready. Of course the 8251 is much simpler. It only responds to two instructions: the 'MODE' and the 'COMMAND'. Both of these are one byte values which you can pass into the control port (with an assembler LD or in BASIC with a POKE). Each of these bytes consist of 8 bits, all of which have their meanings to the 8251, resulting in the awe-inspiring tables in the following figures. Although the tables look a bit daunting at first, the actual programming is quite easy.

Generally speaking, the 'COMMAND' instruction tells the 8251 what to do (i.e; send or receive data), while the 'MODE" instruction says how to do it (i.e. BAUD rate, character length, etc). The actual initialization of the 8251 is accomplished in three steps.

1) Reset the USART. 2) Set the operating mode. 3) Define the operation to be performed.

RESETTING- This is the simplest of the 'COMMAND' instructions. Referring to the figures, you will see that the hex '40' bit is labelled 'IR' (internal reset). Any value which includes this hex '40' bit ON, when POKE'd into the control port will reset the 8251. Once the 8251 has received the reset command it will stop whatever it is doing and wait for a 'MODE' followed by another 'COMMAND' instruction in that order.

MODE SETTING- The next step is to specify the operating mode of the 8251. To do this correctly you have to know the format in which you will be exchanging data with the other device. You will need to know:

1) The character size.

This is the number of bits forming one byte of transmitted data, which can vary from a minimum of 5 to a maximum of 8 bits. From the figures, the levels 'L2' & 'L1' (hex 04 & 02) in combination are used to specify the character length. 7 bits/character would be L2=1 and L1=0, for a value of 4+0=4. 8 bits/character would be L2=1 and L1=1, for a value of 4+2=6.

2) The parity.

Parity generation (if specified) will result in an extra bit being transmitted per character. If the receiving device expects parity bits and you don't send them then it will process every byte it receives as being in error, the same holds true for data which you receive. The reverse is also bad, if it doesn't expect parity and you do. It won't know what to make of the extra bits you send, and you will receive too few.

Parity (or not) is set by the bit labelled 'PEN' (parity enable), a hex '10' in the figure. The type of parity generation (even or odd), is specified by the bit labelled 'EP' (even parity), hex '20'.

To generate even parity, you would set 'PEN' =1 and 'EP' =1 for a value of hex 10 + hex 20 = hex 30. In decimal, that would be 16+32=48. To generate no parity, you would set 'PEN'=0, and the value of 'EP' would be ignored by the 8251.

3) 'STOP' bits:

The number of stop bits is set by the two bits 'S2' & 'S1' in combination. Generally speaking printers seem to ignore extra stop bits; other computers may or may not. The bit values of S2 and S1 are hex 80 and hex 40 respectively; thus the value for two stop bits would be S2=1, S1=1 for a total of hex 80 + hex 40 = hex C0 (faked you out-- thought it would be 120..ed) In decimal, this is 128+64=192.

4)BAUD rate:

The baud rate is specified to the 8251 as a function of the data clock rate. The selection is controlled by the values given for the bits labelled 'B2' and 'B1', hex 02 and hex 01 respectively,. As shown, in the figures, the data clocks are both connected to the output from the clocking circuit (either IC 8 or IC 9 via IC 6), and are pins 9 and 25 of the 8251. Since the transmit and receive data clocks are separate pins on the 8251 it is theoretically possible to transmit and receive data at different rates; in actual practice this would be unusual to say the least.

Assuming that the data clocks are derived from IC 9, the clock rate would be 19.3 kHz (19,3000 cps). From the figures, you have three possible BAUD rates available (using B2 and B1): clock rate divided by 1, by 16, and by 64. The baud rates available in this configuration would therefore be 19.2 KB (very fast, and which I have not tried), 1.2 KB (or 1200 baud), and 300 baud. These would be specified as:

B2=0, B1=1 for 19.2 KB
B2=1, B1=0 for 1200 baud.
B2=1, B1=1 for 300 baud.

Example of MODE Instruction

Let's assume you want to use the following setup: Character length = 7 bits, parity = even, stop bits =1, and baud rate =300.

The MODE instruction value would be calculated as follows:

L2=1. L1=0
PEN=1

EP=1
S2=0, S1=1
B2=1, B1=1

Totaling the above, we get hex 08+10+20+40+03= hex 7B (=decimal 123). Therefore the operating mode above would be set by POKEing the value 123 into the CONTROL port.

The COMMAND Instruction:

Now that the 8251 has been told of the data format, we have to tell it what we expect it to do. This is accomplished by POKEing the COMMAND instruction into the control port. We have already covered the IR (internal reset) command; now let's see what the rest of the command bits are for.

EH (enter hunt) hex 80 is only used in the synchronous mode, and is not applicable to most micro applications.

RTS (request to send) hex 20 will set pin 23 of the 8251 low.

ER (error reset) hex 10 will set the status read error flags (PE, OE, FE) OFF. The 8251 will continue to function after an error conditions occurs, but if you want to keep track of errors it is best to reset the error flags as you go.

SBRK (send break) hex 08 will send a BREAK signal., some mainframe computer terminal programs will recognize this as a QUIT transmission signal.

RxEN (receive enable) hex 04 tells the 8251 that you expect to receive data.

DTR (data terminal ready) hex 02 if on will force pin 24 of the 8251 low.

TxEN (transmit enable) hex 01 tells the 8251 that you will be sending data.

COMMAND Instruction Example

To ready the 8251 to transmit and receive data, you would code the COMMAND instruction as follows:

EH, IR, RTS, SBRK, and DTR all set OFF (0).

ER, RxEN, TxEN would all be set ON. This would give us a value of hex 15 or decimal 21 to POKE into the control port.

Status Read Flags

Now that the 8251 is ready to work you can start sending and receiving data. The problem is, though, how are you to know when a byte has been received, or when it is time to send another? The 8251 will supply the answer to these and many other questions by way of status bits which you obtain by reading (PEEKing) the control port. This operations is just as simple as it sounds; every time you PEEK the control port, you will get a value which is the sum of all the flag bits currently ON. This status read does not interfere in any ongoing operation in which the 8251 is engaged, nor does it change any of the settings you so laboriously set up with the MODE or COMMAND instructions. The various flags are shown in the figures.

DSR (data set ready) hex 80 is set to 1 if the signal at pin 22 of the 8251 goes LOW. On the wiring diagram, you will note that the 1489 pin 4 is tied high. Since the 1489 is an inverter this forces a low condition to pin 22 of the 8251 causing the DSR flag to always be ON. This is the kind of setup to use with a modem since they don't generally use this signal anyway. With some printers, however, this signal is used to indicate that the printer buffer is full, out of paper, etc. In this case do not connect this pin on the 1489 to +5 as shown, but leave it connected only to the printer through the RS-232 cable. This way the printer signal will be reflected through the 1489 to the 8251 to alert your program to stop sending data. The DSR bit will normally go ON when the printer cannot accept data.

SYNDET (break received) hex 40 will not normally be on, since it is unlikely your micro would receive a BREAK signal.

FE (framing error) hex 20 is set if the 8251 did not receive a valid STOP bit following a data character.

OE (overrun error) hex 10 is set if a second character was received before your program accepted the preceding one. The 8251 will continue working, but you will have lost the preceding data byte.

PE (parity error) hex 08 is set if the 8251 detects a parity error on a received character. This will not inhibit the operation of the 8251, but you can assume that the character received is bad. This can also be caused by not having the same parity generation and checking at both ends of the transmission.

TxMT (transmit buffer empty) hex 04 indicates that the 8251 transmit buffer

is empty (ready to accept another byte to send).

RxRDY (receiver ready) hex 02 indicates that a character is ready to be accepted by the CPU.

Programming the 8251

As can be seen by the foregoing, the basic steps in a terminal program would be:

1) Initialize the 8251 to send and receive. 2) Check the status flags for RxRDY; if it is ON, accept, translate and display the character. 3) Check the keyboard for activity. Translate and send it if TxMT and TxRDY flags are ON. 4) Keep on looping through steps 2 and 3 until some QUIT condition is hit.

Following is a sample BASIC program showing the steps above. I left it deliberately simplistic; it will read and display a message from the modem, then turn around and accept and send a message from the keyboard. It then loops back to read the modem, and on. and on.

Lines 10 and 20 assign labels to the addresses of the control port. (C is 8193) and the data port (D is 8192).

Line 30 assigns a label to the value of the mode set instruction (M = 123; 7 bits/char, even parity, 300 baud, and 1 stop bit).

Line 40 assigns a label to the status flags 'ready to send.

Lines 100 through 130 initialize the 8251; first the reset command (line 110), then the mode instruction followed by the command.

Lines 200 through 250 read from the mode. Line 220 loops until RxRDY comes on, then a character is collected into X$. When carriage return is detected (13), the routine exits.

Lines 310 through 360 translate (roughly) the ASCII data to Sinclair format. For simplicity, this routine will only accept alphanumerics. (0 through 9, and A through Z).

Line 370 prints the translated character(s) on the screen.

Once the incoming message had been displayed, the program turns around and reads from the keyboard (Lines 410 and 420).

Lines 510 through 530 then set the 8251 into send mode, and lines 610 through 670 translate from Sinclair format into ASCII.

Lines 700 through 720 send your message, checking the status of the 8251 in line 700.

When the messages is sent, lines 800 onward send a carriage return, then go back to the data read routine

For the machine code enthusiast, the same principles apply, except that with the increased speed of execution more elegant routines can be used. The USART can be initialized for both input and output so that you can be sending while data is being received. Translating of the data between Sinclair format and ASCII can also be considerably enhanced, allowing upper and lower case alpha and special characters.

```
  5 REM  ** SET UP LABELS
 10 LET C=8193
 20 LET D=C-1
 30 LET M=123
 40 LET OK=5
 95 REM  ** INIT TO RECEIVE
100 FAST
110 POKE C,64
120 POKE C,M
130 POKE C,20
195 REM  ** READ FROM MODEM
200 LET X$=""
220 IF PEEK C=OK THEN GOTO 210
225 REM  ** MOVE DATA TO X$
230 LET X$=X$+CHR$ (PEEK D)
240 IF PEEK D=13 THEN GOTO 300
250 GOTO 220
305 REM  ** XLATE MSG
310 FOR P=1 TO LEN X$
330 LET Y=CODE X$(P)
335 IF Y>90 THEN LET Y=Y-32
340 LET Y=Y-27
350 IF Y<31 THEN LET Y=Y+7
360 IF Y<20 THEN LET Y=0
365 REM  ** PRINT EACH CHAR
370 PRINT CHR$ Y;
380 NEXT P
390 PRINT
405 REM  *** GET MSG TO SEND
410 INPUT X$
420 CLS
505 REM  *** INIT TO SEND
510 POKE C,64
520 POKE C,M
530 POKE C,17
540 SLOW
605 REM  *** SEND MSG
610 FOR Q=1 TO LEN X$
620 LET Y$=X$(Q)
630 LET Y=CODE Y$
635 REM  *** XLATE TO ASCII
640 IF Y$<"A" THEN LET Y=Y+20
650 IF Y$>"9" THEN LET Y=Y+27
660 IF Y$=" " THEN LET Y=32
670 IF Y$="." THEN LET Y=46
690 REM  *** SEND CHAR
700 IF PEEK C<>OK THEN GOTO 700
710 POKE D,Y
720 NEXT Q
795 REM  *** SEND CARRIAGE RTN
800 IF PEEK C<>OK THEN GOTO 800
```

```
810 POKE D,13
815 REM   *** NOW READ AGAIN
820 IF PEEK C<>OK THEN GOTO 820
830 GOTO 100
```

PUBLISHER'S NOTES

I have heard from several sources via computer and letter that the Midwest Timex-Sinclair Computer Fest was a big success. There were lots of vendors (about 35) and pre-registered individuals. The many motel rooms were all sold out. The fair paid for itself before it was held. The success of this fest for an "orphaned" computer is a remarkable tribute to people who are still fascinated with this machine. The success will guarantee future gatherings of this sort in other locations as well. Frank Davis and his helpers are to be congratulated for a good idea well excecuted.

Sinclair Reseach (in Great Britain) has been sold and the QL distribution in the USA changed again. The current distributer is A+ Computer Response.

We received an interesting letter from N9BBN, John. He indicated that he was considering getting the Comlink I (see ad this issue) RS232 interface and then writing some packet radio software for our Sinclair computers. He talked of putting in split screen Tx-Rx, canned messages, set up for different interfaces and data rates, and standard ARRL messages. John would welcome suggestions on features to be included. (John Mc Ginley, 203 Milwaukee, Madison, WI 53948)

SPLIT SCREEN SCROLL ROUTINE
Dallas Williams WA0MRG

This short routine allows a split screen display with some programs on the ZX81. In essence what it does is set up the entire screen for display use (all 24 lines!). By doing that and then printing what is desired in the lower part of the screen, the upper part can be scrolled independently. I find it works quite well for RTTY, allowing the lower part of the screen to contain the program menu while the upper part of the display scrolls the RTTY data as normal.

I have not tried this routine except in one of my RTTY programs. It works very well there and, with careful application and modification, should work well in some other programs. Note that the system may crash if the routine is not used or placed correctly within the program. Memory location 16418 is part of the display file containing the number of lines in the lower part of the screen. Normally the system boots 2 to this location.

Note also that the bottom part of the display is not totally protected from program control. CLS, for instance, still clears the whole screen. Still, a partial screen scroll is rather handy to have.

First,

POKE 16418 (Allows printing at the bottom of the bottom of the screen that normally is not available.)

At this point, that which is to be printed on the bottom of the screen should be generated and placed using PRINT AT .... From this point on, avoid the use of CLS or data in the lower part of the screen will be lost.

Next,

POKE 16418,X (Eliminates X number of lines at the bottom of the screen from program SCROLL control.)

Now your program can generate the scrolling display you want for the rest of the screen.

That's it. Simple but words great on my RTTY program.

BITS AND BYTES

71. Sometimes there is an RF hot spot on the left side of the T/S 2068 near the TV modulator. If you are having trouble with consistent SAVE's, try moving the tape to the right side. 72. To temporarily delete a line or to provide easily changed options in a long BASIC program, place a quote ' or a REM after the line number. This converts the line into a REM statement and removes its effect from the program without erasing it. 73. From LU8MAD comes the hint to use the following short program to SAVE or LIST a program which does not normally SAVE or LIST. Enter the following:

FAST
ENTER
RAND USR 837
ENTER

Put the tape in your tape deck and turn it on. The difficult tape will be loaded

in such a way that it can now be SAVEd and LISTed.

PRODUCTS OF SPECIAL INTEREST

Here are some more or less new products which are of special interest.

A 512K RAM plug-in for the QL. This is really needed if you make much use of the software which comes with the computer. That software is memory intensive, i.e when loaded you do not have much memory left. From G. Russell Electronics, RD 1 Box 539, Centre Hall, PA 16828, tele (814) 364-1325 for $179.

ZX 81 technical information. From Howard Sams & Co., 4300 W. 62nd St., Indianapolis, IN 46268, tele (800) 428-7267 you can get a Computerfact package which contains technical information designed for servicing these computers. $20.

T/S 2068 technical information. From Time Designs, 29722 Hult Rd., Colton, OR 97017 you can get a revised and re-edited T/S 2068 technical manual. $25.

QZX FRIENDS

VE2LI -- George now has three computers, a ZX81, a T/S 2068, and a Commodore 64. He recently renewed and stated that he had to pay Canadian $17 for the US $12 money order to do so.

WA0ZTU -- Rolf is a new subscriber with a ZX81 and 64K RAM. He has an external keyboard with homebrew circuits for 10 dedicated auto-shift keys, a repeat key, and a shift lock function.

NR5A -- Jerry was one of a surprising numbers who asked for the detailed plans for the mechanical work necessary to add a TI keyboard. he also indicated his intention to put his T/S 1000 on cw and RTTY.

W9JEX -- Andrew said he has three ZX81 - T/S 1000 with a 64K memory from Memotech. He would like more programs for these computers. He says that he does not intend to purchase a more sophisticated computer soon, since he has much to learn for his ZX-81.

KD6TO -- Dave says the only thing his Sinclair lacks is packet radio.

KA4FRF -- Tom writes from Karachi that he enjoyed the Tom Woods article on interrupts. There is not too much written about their use that he has been able to find. He has some interrupt driven trace and clock programs that work on the 2068 with a little fix-up of the ROM calls. He says that the Russell ROMFIX was probably the cheapest fix that he has purchased that did so much: Trace, Clocks, and Sprites plus an "interrupt manager" that will run more than one interrupt driven routine.

N9BBN -- John wrote, "Sorry I'm late (with renewal). Please make sure I get the March 86 issue - I really want to get my T/S 1000 on packet!

K6BPT --Skip said he saw a notice about the QZX FIDO BBS on packet a KD6SQ. He also says he uses his and T/S 1000 for ham programs. used to write commercial financial programs. He did not do ham programs because KA2LHO was doing such a great job.

WD0DCU -- Anthony says he has no intention to switch to the QL. He is having too much fun with the T/S 100 and 2068.

Edward Radtke visited us in person this month. He told us of his efforts to keep his users group in Louisville, Ky active. He also told us about his collection of Sinclair products including a watch kit. See his unclassified ad in this issue as he attempts to complete his computer collection.

KQ2F -- Bob sent us a nice long letter and enclosed three photographs of a super keyboard and enclosure for his ZX81. It includes a stringy floppy tape drive, built in expansion port, led's, and lots of switches. He uses it for satellite tracking and other dedicated HAM uses. He says that he has turned over the NCS job for the 7.240 Sunday QZX net to WB4DLU, Charlie.

UNCLASSIFIED ADS

WANTED: UNCLASSIFIED ADS for publication in QZX. Cost is 1 cent for each keystroke (characters, blanks, etc.). Payment must be included with order. This is an excellent way to sell your program or unwanted equipment, ask for help, or find a specific program. Send your ad with payment to QZX, 2025 O'Donnell, Las Cruces, NM 88001.

FOR SALE: 9 late issues SYNC, Spring SQ. $15+shipping. Gunther, 1402 W. Craig, Moses Lake, WA 98837.

COMLINK I will link your ZX81 or TS1000 to the outside world. With this easy to use adapter and any 300 baud modem you can:

* Access databases such as Compuserve, The Source, or Dow Jones News/Retrieval.

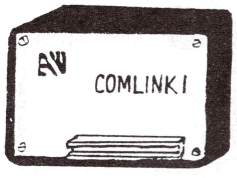
* Use your school or company mainframe computer from home.

* Use with a Terminal Node Controller (TNC) for packet radio.

* Print information received using the TS printer.

All software is included in EPROM so there is nothing to load before using COMLINK I. All operating power is supplied by the Sinclair. Simply plug the unit into your Sinclair and connect it to an RS-232C compatible modem or TNC. One 'USR' call gets you started!

Each unit is attractively cased, fully tested, and is sent with complete operating instructions. Only $54.95 includes shipping.



To order, send check or money order to:

A. Eckhardt
918 Anna Street
Boalsburg, PA. 16827

***** QUANTUM COMPUTING PACKAGES From QCOM *****
Phone or Send QCOM A CQ CALL For YOUR QL!
Act Now while Introductary Prices & Supplies Remain Available!!

The SINCLAIR QL Computing System: {MC68000 Breed} 32 Bit Programming Archetecture With SuperBASIC/QDOS Coprocessing Environment.
Currently Retails Elsewhere from $299. to $399. Your Price: $288.00
QL HI-RES Color Monitor: Reatils for $400.00 Your Price: $340.00
QL Custom Matched Printer: Retails for $349.00 Your Price: $275.00
Monitor & RS 232 Cables: $15.00 Ea.

Software For Your QL: (Prices Average or Exceed 20% Below Retail!)
Word Processor, Spreadsheet, Database, Graphics Facility $00.00 Included FREE!
QL ASSEMBLER $49.95
QL MONITOR $34.95
QL TYPING TUDOR $34.95
QL HOME FINANCE MANAGER $34.95
QL CAVERN (GAME) $27.95
QL BOUNDER (GAME) $34.95
QL PAWN (CHESS) $49.95
QL DECISION MAKER (EXC. Business Evaluation Tool) $49.95
QL PROJECT PLANNER (EXC. Tracking/Management) $49.95
QL ENTREPRENEUER (EXC. Venture Decision Facility) $49.95
QL GARDENER (Garden & Farm Management Program) $34.95
The QMONIX Line of Business, Science, and QL Configuration Software (Write for Info!)

NOTE: PLEASE ADD 5% To Your ORDER TOTAL to Cover the Postage & Handling. Money Orders, PAYMENTS VIA: Money Orders, Bank Certified Checks or Western Union Wire Always Ship SAME DAY Recieved. Personal Checks Ship After Clearing. Orders sent in without P&H are held & billed for such. Please Comply with these requirements so WE can Remain TOPS IN SERVICE TO YOU! QL Questions?? CALL: (1201) 328-8846
Make Orders Payable to: QUANTUM COMPUTING.

ADDRESS ORDERS TO: QUANTUM COMPUTING, PO BOX 1280, DOVER, N.J. 07801

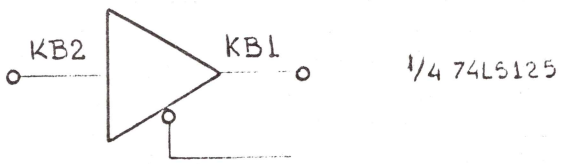
USING EXTRA KEYS ON TI KEYBOARD

The MAY 85 and FEB 86 issues of QZX described how to wire a TI 99/4A surplus keyboard onto your TS1000 (and kismans).

The keyboard have six more key than you need.

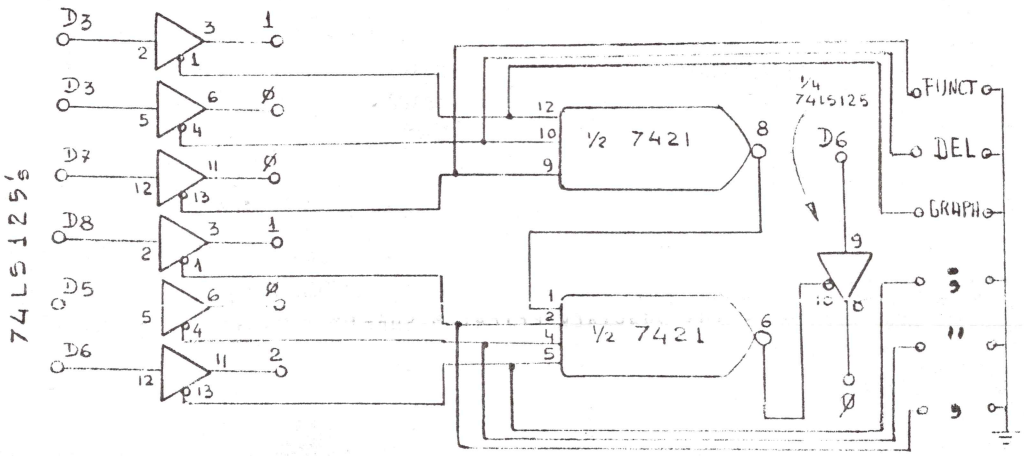
You can wire these six keys in Parallel with existing keys to access shifted functions without pressing the SHIFT key. Use ICs to implement the Permanently shifted keys.

You must choose six keys according to needs, in the TABLE 1. Then replace with you elected values of KB2 & KB1 in the circuit as following drawing.



I have chosen [.] ["] [,] [DEL], [FUNCT] & [GRAPH] key for my keyboard.

The circuit following shows six wired keys.



CONSTRUCTION

a)PARTS

-one Dual 4 In AND Gate (7421)
-two Triestate Quad 2-In Buffer (74LS125)
-3 14 Pin DIP sockets (solder tail type)
-insulated wire, Etc.

b)WIRE THE CIRCUITS

Following the circuits, make the appropiate connection to IC's. This is best accomplished by making them directly to the DIP socket. For connections that will be made to the keyboard, solder small Pieces of wire to the appropiate Pins of the sockets.

Then use the wire list in TABLE of Page 11 of FEB'86 QZX issue, to connect insulated wire (30-AWG) from the keyboard to the DIP sockets.

Isolate means to cut traces away from switch terminal Pad correspondent to the six extra keys hitherto useless.

Connected means of insulated wire an terminal of each those keys and grounded. The other terminal of each keys is connected to the appropiate Pins of DIP sockets (see circuit).

c)AFFIX

Put the chips in the DIP sockets, making certain that it in right side.

Affix the IC's assembly to the keyboard or console, upside down, using double-side tape or a small dab of rubber cement RTV.

Relabel your new keys. Unfortunately actual label is very difficult to erase. Dry rub-down letters make Professional-looking labels for key tops, and a coat of lacquer protects those from wear.

TABLE 1

KEY NAME	KB2	KB1	KEY NAME	KB2	KB1
EDIT	D1	0	AND	D1	1
THEN	D1	2	TO	D1	3
←	D1	4	↓	D3	4
↑	D3	3	→	D3	2
GRAPH	D3	1	DEL	D3	0
"	D2	0	OR	D2	1
STEP	D2	2	<=	D2	3
<>	D2	4	>=	D5	4
$	D5	3	(	D5	2
)	D5	1	"	D5	0
STOP	D4	0	LPRINT	D4	1
SLOW	D4	2	FAST	D4	3
LLIST	D4	4	**	D7	4
-	D7	3	+	D7	2
=	D7	1	FUNCT	D7	0
:	D6	1	;	D6	2
?	D6	3	/	D6	4
*	D8	4	<	D8	3
>	D8	2	,	D8	1

Figure 5.
RS232 Standard for 25 pin ""D'' connector plug.
MALE plug to modem

PIN	USE (USART)	CONNECT TO
3	Receive data	Pin 1 on 1489
2	Transmit data	Pin 11 on 1488
1	ground	CPU ground
0	ground	CPU ground
FEMALE plug to printer.		
3	Transmit data	Pin 11 on 1488
20	Printer Ready	Pin 13 on 1489
1	ground	CPU ground
7	ground	CPU ground

Figure 3.
The ""COMMAND'' Instruction Format

80	40	20	10	08	04	02	01	Bit Value (hex)
EH	IR	RTS	ER	SBRK	RxEN	DTR	TxEN	Label

Bit MEANINGS
EH : not used (synchronous mode only).
IR : Internal Reset.
1 = reset the USART.
RTS : Request To Send control.
(not required)
ER : Error Reset.
1 = reset error status flag.
SBRK : Send BReaK.
(pretty exotic)
RxEN : Receive ENable
1 = get ready to receive data
DTR : Data Terminal Ready control.
(not required)
TxEN : Transmit ENable.
1 = get ready to send data.

Figure 2.
The ""MODE'' Set Instruction Format

80	40	20	10	08	04	02	01	Bit Value (hex)
S2	S1	EP	PEN	L2	L1	B2	B1	Label

S2	S1	MEANING (""STOP'' bits)
0	0	Invalid
0	1	1 STOP bit
1	0	1 1/s12 STOP bits
1	1	2 STOP bits

L2	L1	MEANING (character length)
0	0	5 bits/slchar
0	1	6 bits/slchar
1	0	7 bits/slchar
1	1	8 bits/slchar

PEN (Parity ENable)
0 disable:
no parity generation on output
no parity checking on input
1 enable:
Parity checking on input
Parity generation on output,
EP (Even Parity)
0 Even parity
1 Odd parity
(Active only if ""PEN'' = 1)

B2	B1	MEANING (BAUD Rate as a function of the data clock rate)
0	0	Invalid
0	1	data clock /sl 1
1	0	data clock /sl 16
1	1	data clock /sl 64

Figure 4.
The 8251 Status Flags

80	40	20	10	08	04	02	01	Bit Value (hex)
DSR	SYNDET	FE	OE	PE	TxMT	RxRDY	TxRDY	Label

Bit Meanings:
DSR : Data Set Ready.
(only used by some printers)
SYNDET : Break received.
1 = break was received.
FE : Framing Error.
1 = invalid character received.
(probably timing or wrong BAUD rate).
OE : Overrun Error.
1 = DATA lost: your program was
too slow to accept a byte.
PE : Parity Error.
1 = xmsn error, or wrong parity.
TxMT : Transmitter eMpTy.
1 = ready to send another character.
RxRDY : Receiver ReaDY.
1 = received byte ready for CPU.
TxRDY : Transmitter ReaDY.
1 = Transmit buffer can accept
another byte to send.

THIS PROGRAM BY BOB HOWARD, WA6DLI IS USEFUL IF YOU WISH TO
 EVALUATE A MONITOR OR TV FOR USE WITH YOUR 2068.
BECAUSE IT HAS A ROUTIME TO PRINT "m" ON THE SCREEN IN BOTH 32
 AND 64 COLUMN MODE YOU CAN SEE HOW GOOD (OR BAD) YOUR TV OR
 MONITOR WILL BE.
PERHAPS YOU COULD TAKE YOUR COMPUTER AND THIS PROGRAM ON
 CASSETTE TO CHECK ON A MONITOR BEFORE YOU BUY IT! YOU WOULD
HAVE TO HAVE A COMPATIBLE CABLE WITH YOU SO SHOULD MAKE IT UP
BEFOREHAND.

```
  1 CLEAR 63199
  2 FOR I=63200 TO 63245
  3 READ A
  4 POKE I,A
  5 NEXT I
  6 DATA 62,128,24,1,175,245,62,128,211,255,62,1,211,244,241,20
5,142,14,50,194,92,175,211,244,167,33,0,64,17,0,96,1,0,27,48,1,2
35,58,194,92,254,0,200,237,176,201
  7 CLS : FOR N=0 TO 703: PRINT "m";: NEXT n
  8 RANDOMIZE USR 63200
  9 RANDOMIZE USR 63244
 10 CLS
 20 REM FOR SINCLAIR SPECTRUM (ORIG. NOT FOR TS2068)
 30 REM BY NICK FOOT G8MCQ
 40 PAPER 0: BORDER 0: INK 7: CLS
 50 PRINT "1) RED RASTER"
 60 PRINT "2) GREEN RASTER"
 70 PRINT "3) BLUE RASTER"
 75 PRINT "4) WHITE RASTER"
 80 PRINT "5) CROSS HATCH"
 90 PRINT "6) COLOUR BARS"
 95 PRINT "7) RESOLUTION"
 97 PRINT " & 64 COLUMN RESOLUTION"
100 PRINT
110 PRINT
120 PRINT "INPUT NUMBER OF TEST ROUTINE"
122 PRINT ''"Adapted from TV Test program by"''"Nick Foot G8MCQ
in Dec '82"''"Supplement to Practical Wireless"''"TS2068 Version b
y WA6DLI"''"Hi Res from H.Weppler CATS 5/86"
130 IF INKEY$="" THEN  GO TO 130
140 IF INKEY$="1" THEN  GO TO 200
150 IF INKEY$="2" THEN  GO TO 250
155 IF INKEY$="3" THEN  GO TO 300
160 IF INKEY$="4" THEN  GO TO 330
170 IF INKEY$="5" THEN  GO TO 350
185 IF INKEY$="6" THEN  GO TO 500
188 IF INKEY$="7" THEN  GO TO 600
190 GO TO 130
200 REM RED RASTER
210 BORDER 2: PAPER 2: CLS : PAPER 0: GO TO 40
250 REM GREEN RASTER
260 BORDER 4: PAPER 4: CLS : PAPER 0: GO TO 40
300 REM BLUE RASTER
310 BORDER 1: PAPER 1: CLS : PAPER 0: GO TO 40
330 REM WHITE RASTER
340 BORDER 7: PAPER 7: CLS : PAPER 0: GO TO 40
350 REM CROSS HATCH
355 CLS
360 FOR N=8 TO 176 STEP 20
370 PLOT 8,N
380 DRAW 240,0
390 NEXT N
400 FOR N=8 TO 255 STEP 20
410 PLOT N,8
420 DRAW 0,160
430 NEXT N
440 CIRCLE 256/2,176/2,75
450 PAUSE 0: GO TO 40
505 CLS
520 FOR I=0 TO 21
530 FOR N=1 TO 7
540 PRINT  INK N;"????"; : REM ? IS SHIFT GRAPHIC 8
550 NEXT N
555 PRINT
560 NEXT I
570 PAUSE 0: GO TO 40
600 REM RESOLUTION TEST
605 CLS
610 FOR N=0 TO 703
620 PRINT "m";
630 NEXT n
635 PAUSE 200: INK 0: PAPER 7: BORDER 7: CLS
636 FOR n=0 TO 703
637 PRINT "m";
638 NEXT n
640 PAUSE 200: GO TO 700
700 OUT 255,6: PAUSE 0: OUT 255,128
702 RANDOMIZE USR 63204: GO TO 705
705 PAUSE 200: CLS : RUN
```

73, W4GHV & K5XY

QZX presents the best software & hardware projects available for the radio amateur. While the contents are believed to be accurate, no warranty is extended on any program or project. No investigation has been made of the claims or reputation of any advertiser. It would be greatly appreciated if all correspondence requiring an answer would be accompanied by a self-addressed stamped envelope (SASE). Copyright by QZX, 1986. All rights reserved, photocopying by permission only.

****
Please send me QZX for 1 year, starting
immediately. Enclosed is $12.00 for twelve issues. (Those with addresses not in Canada, the US, or Mexico send $25.00 in US funds for air mail delivery.. Thanks (04/86)
*
* Name _____ *
* Call _____ *
*
* Address _____ *
*
* City _____ *
*
* State _____ ZIP _____ *
****

QZX Journal
2025 O'Donnell Drive
Las Cruces, N.M.
88001

BULK RATE
U.S. POSTAGE
PAID
LAS CRUCES, N.M.
PERMIT NO. 655

86/09

W*9MAT
KEN MATZNER
RR 5 BOX 15
CHARLESTON, IL 61920